Semantic Image Segmentation with Contextual Hierarchical Models

Mojtaba Seyedhosseini and Tolga Tasdizen, *Senior Member, IEEE*

Abstract—Semantic segmentation is the problem of assigning an object label to each pixel. It unifies the image segmentation and object recognition problems. The importance of using contextual information in semantic segmentation frameworks has been widely realized in the field. We propose a contextual framework, called *contextual hierarchical model (CHM)*, which learns contextual information in a hierarchical framework for semantic segmentation. At each level of the hierarchy, a classifier is trained based on downsampled input images and outputs of previous levels. Our model then incorporates the resulting multi-resolution contextual information into a classifier to segment the input image at original resolution. This training strategy allows for optimization of a joint posterior probability at multiple resolutions through the hierarchy. Contextual hierarchical model is purely based on the input image patches and does not make use of any fragments or shape examples. Hence, it is applicable to a variety of problems such as object segmentation and edge detection. We demonstrate that CHM performs at par with state-of-the-art on Stanford background and Weizmann horse datasets. It also outperforms state-of-the-art edge detection methods on NYU depth dataset and achieves state-of-the-art on Berkeley segmentation dataset (BSDS 500).

Index Terms—Semantic segmentation, image segmentation, edge detection, hierarchical models, membrane detection, connectome

1 Introduction

Semantic segmentation is of substantial importance for a wide range of applications in computer vision [1]. It is the primary step towards image understanding and integrates detection and segmentation in a single framework [2]. For instance, in a dataset of horse images, semantic segmentation can be thought of as the task of labeling each pixel as part of a horse or non-horse, i.e., background. In more complicated cases such as outdoor scene images, it might require multiple labels, e.g., buildings, cars, roads, sky etc. This general definition can also be extended to the edge detection problem where each pixel is classified as edge or non-edge in a binary-decision framework.

Pixels can not be labeled based only on a small region around them. For example, it is almost impossible to distinguish a pixel belonging to sky from a pixel belonging to sea by only looking at a small patch around them. Therefore, a semantic segmentation framework needs to take into account short-range and long-range contextual information. Contextual information has been widely used for solving high-level vision problems in computer vision [3], [4], [5], [6]. Contextual information can refer to either inter-object configuration, e.g., a segmented horse's body may suggest the position of its legs [3], or intra-object dependencies, e.g., the existence of a keyboard in an image implies that there is very likely a mouse near it [4]. From the Bayesian point of view, contextual information can be interpreted as the probability image map of an object, which carries prior information in the maximum aposteriori (MAP) pixel classification problem.

An important question about any semantic segmentation method is how it takes contextual information into account. The main challenge is to pool contextual information from a large neighborhood while keeping the complexity tractable [2]. A common approach is to use a series of cascaded classifiers [3], [5], [6], [7]. In this architecture, each classifier is sequentially trained using the outputs of the previous classifiers as inputs. This gradually increases the area of influence and allows later classifiers in the series to obtain contextual information from larger neighborhood areas. However, they have a drawback that they do not obtain contextual information at multiple scales. Multi-scale processing of images has been proven critical in many computer vision tasks [8], [9]. OWT-UCM [10] takes advantage of processing the input image at multiple scales through a hierarchy. This leads to state-of-the-art performance for edge detection applications. Farabet et al. [2] showed that using multi-scale convolutional networks (ConvNets) can improve the performance of ConvNets dramatically for semantic segmentation.

This paper presents a contextual hierarchical model (CHM), which is able to obtain contextual information at multiple resolutions. Similar to cascaded classifier models, CHM learns a series of classifiers consecutively, but unlike those models, it trains classifiers at multiple resolutions in a hierarchy. The main advantage of CHM is that it targets a posterior probability at multiple resolutions and maximizes it greedily through the hierarchy. This allows CHM to cover a large contextual window without adding intractable complexity. While common approaches to semantic segmentation usually need postprocessing to ensure the consistency of labels, the use of a large contextual window reduces the requirement for sophisticated postprocessing methods.

• *The authors are with the Electrical and Computer Engineering Department and the Scientific Computing and Imaging Institute, University of Utah, Salt Lake City, UT 84112. E-mail: {mseyed, tolga}@sci.utah.edu.*

Manuscript received 15 Jan. 2015; revised 28 July 2015; accepted 16 Aug. 2015. Date of publication 26 Aug. 2015; date of current version 8 Apr. 2016. Recommended for acceptance by V. Ferrari.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPAMI.2015.2473846

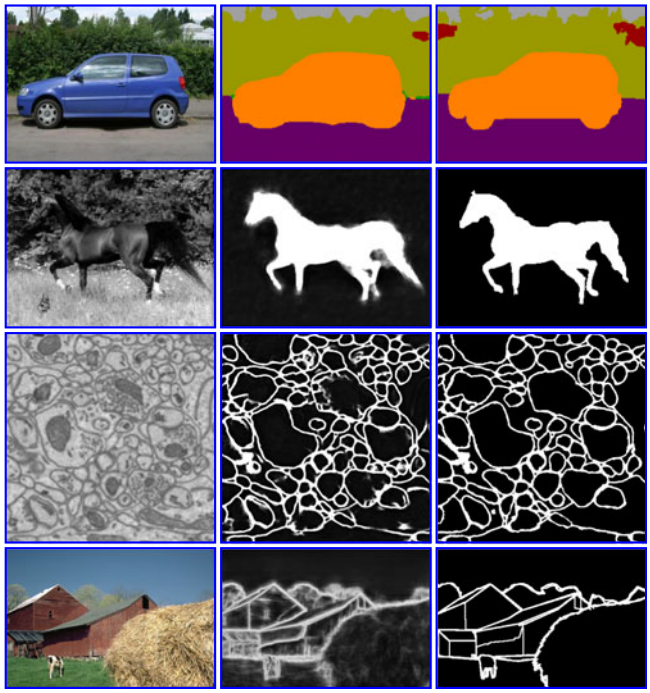

Fig. 1. Results of CHM on different tasks. *First row:* Semantic segmentation (Stanford background dataset [18]. *Second row:* Horse segmentation (Weizmann dataset [17]. *Third row:* Membrane detection (mouse neuropil dataset [9]). *Fourth row:* Edge detection (Berkeley dataset [19]). See Section 4 for details.

A striking characteristic of our proposed method is that it is purely based on input image patches and does not make use of any shape fragments or object models, therefore, it is applicable to a wide range of applications such as edge detection and image labeling. While some approaches such as [10], [11], [12], [13] can only be applied to edge detection problems and other approaches such as [14], [15], [16] are only designed for the image labeling problem, CHM can handle both problems equally well without any modification.

In extensive experiments, we demonstrate the performance of CHM on a couple of challenging vision tasks: Horse segmentation in the Weizmann dataset [17], outdoor scene labeling in the Stanford background [18]. We also show the performance of CHM for edge detection on the popular BSDS 500 [19] and NYU Depth (v2) [20] datasets. In all cases, CHM results in either state-of-the-art or near state-of-the-art performance. In addition, we apply CHM on two electron microscopy (EM) datasets for cell membrane detection (Drosophila VNC [21], [22] and mouse neuropil [9]). CHM outperforms many existing algorithms for membrane detection and can be used as the first step towards reconstruction of the connectome, i.e., the map of neural connectivity in the mammalian nervous system [23]. Some samples of CHM results are shown in Fig. 1.

An early version of this work was first presented in [24]. This journal version reports more comprehensive experiments and gives more theoretical insight into CHM.

2 Related Work

2.1 Graphical Models

There have been many methods that employ graphical models to take advantage of contextual information for semantic segmentation. Markov random fields (MRF) [18], [25], [26], [27] and conditional random fields (CRF) [28], [29] are the most popular approaches. He et al. [28] used CRF to capture contextual information at multiple scales. Larlus and Jurie [25] used MRF on top of a bag-of-words based object model to ensure consistency of labeling. Gould et al. [18] defined an energy function over scene appearance and geometry and then developed an efficient inference technique for MRFs to minimize that energy. Kumar and Koller [26] formulated the energy minimization as an integer programming problem and proposed a linear programming relaxation to solve it. Koltun [30] proposed an efficient approximate inference method for dense CRFs defined over pairwise pixels. Yao et al. [31] formulated the holistic scene understanding problem as a structure prediction in a graphical model. Tighe and Lazebnik [27] proposed an MRF-based superpixel matching that can be easily scaled to large datasets. Ladicky et al. [29] introduced a hierarchical CRF, which is able to combine features extracted from pixels and segments. For inference, they used a graph-cut [32] based method to find the MAP solution. Ren et al. [16] used a superpixel MRF together with a segmentation tree for RGB-D semantic segmentation.

Many of the graphical model methods rely on presegmentation to superpixels [16], [27] or multiple segment candidates [26], [33]. More powerful region-based features can be extracted from superpixels compared to pixels. Moreover, presegmentation to superpixels improves the computational efficiency of these models. However, it is known that superpixels might not adhere to the image boundaries [34] and thus can decrease labeling accuracy [16]. This motivated approaches using multiple segments as hypothesis. However, these methods can be problematic when dealing with cluttered images [29]. This motivated methods with hierarchical segmentation [29], [35].

Unlike previously cited approaches, our proposed method does not make use of any presegmentations or exemplars and works directly on image pixels. This allows our model to be applied to different problems without any modifications. Moreover, inference is simpler in our CHM compared to graphical models. It only requires the evaluation of classifier function and does not require searching the label space as in CRFs [36].

2.2 Convolutional Networks

Deep learning is a very active area of research and has been widely used in the computer vision field. Convolutional networks (ConvNet) [37] are one of the most popular deep architectures. They were initially proposed for character recognition [37], but later applied successfully to image classification [38], [39] and object detection [40], [41]. They have also been used for biological image segmentation [42], [43], [44] and semantic segmentation [2], [36], [45], [46], [47]. Jain et al. [42], Turaga et al. [43], and Ciresan et al. [44] used convnets for membrane detection and cell segmentation in EM images. Grangier et al. [36] trained a ConvNet by iteratively adding new layers for scene parsing. Farabet et al. [2] proposed a multi-scale ConvNet for scene parsing. Their framework contains multiple copies of a single network which are applied to a scale-space pyramid of input images. They performed some postprocessing methods to clean up

the outputs generated by the ConvNet. Zheng et al. [45] formulated CRFs as recurrent neural networks and built a deep network, which leverages the benefits of CRFs and convolutional networks for semantic image segmentation. Chen et al. [46] also combined a fully connected CRF with deep convolutional networks to improve the localization in semantic segmentation. Finally, Long et al. [47] employed a fully convolutional network, which can efficiently handle dense prediction tasks like semantic segmentation.

ConvNets can cover a large contextual area compared to other methods, but they need several hidden layers with many free parameters. Training the ConvNets is computationally expensive and might take months or even years on CPUs [44]. Hence, GPU implementations, which speed up the training process, are usually needed in practice. Unlike ConvNets, our CHM can be trained on CPUs in a reasonable time due to its stage by stage training process. In the experiments we show the performance of CHM in comparison with the ConvNets proposed in [2], [42], [44], [47].

2.3 Cascaded Classifiers

The idea of using multiple classifiers to model context has been proven successful to solve different computer vision problems. Fink and Perona [48] proposed the mutual boosting framework which takes advantage of multiple detectors in a boosting architecture for object detection. Torralba et al. [4] proposed the boosted random field (BRF), which uses boosting to learn the graph structure of CRFs, for object detection and segmentation. Heitz et al. [5] proposed a different architecture to combine multiple classifiers, called cascaded classifier model, for holistic scene understanding. Li et al. [6] introduced a feedback enabled cascaded classification model which jointly optimizes several subtasks in a two-layer cascade of classifiers. In a more related work, Tu and Bai [3] introduced the auto-context algorithm, which integrates both image features and contextual information to learn a series of classifiers, for image segmentation. A filter bank is used to extract the image features and the output of each classifier is used as the contextual information for the next classifier in the series. Jurrus et al. [7] also trained a series of artificial neural networks (ANN) [49], which learns a set of convolutional filters from the data instead of applying fixed filter banks to the input image. Their series architecture was improved by employing a multi-scale representation of context during training [50]. The advantage of the cascaded classifier model over ConvNets is its easier training due to treating each classifier in the series one at a time.

We also introduce a segmentation framework that takes advantage of both input image features and contextual information. Similar to the auto-context algorithm, we use a filter bank to extract input image features. But we use a hierarchical architecture to capture contextual information at different resolutions. Moreover, this multi-resolution contextual information is learned in a supervised framework, which makes it more discriminative compared to the above-mentioned methods. From the Bayesian point of view, CHM optimizes a joint posterior probability at multiple resolutions simultaneously. To our knowledge, supervised multi-resolution contextual information has not previously been used in a semantic segmentation framework.

2.4 Edge Detection

There is a large body of work in the area of edge detection. Many unsupervised techniques have been proposed for edge detection [10], [51], [52], [53]. The Canny edge detector [51] is one of the earliest and gPb [53] is one of the latest among these approaches. More recently, supervised techniques have been explored to improve the edge detection performance [11], [12], [54], [55], [56], [57], [58]. Martin et al. [54] and Dollár et al. [55] used a classifier on top of extracted features to find edges.

Mairal et al. [56] proposed to learn discriminative sparse dictionaries to distinguish between "patches centered on an edge pixel" and "patches centered on a non-edge pixel". Ren and Bo [12] used gradients over learned sparse codes instead of hand designed gradients of [54] to achieve state-of-the-art performance. Lim et al. [58] defined a set of sketch tokens by clustering the patches extracted from groundtruth images. Then, they trained a random forest (RF) to detect those tokens at test time. Finally, Dollár and Zitnick [11] made use of different edge patterns, e.g., T-junctions and Y-junctions, present in images, and used a structured random forest to learn those patterns. Their method is fast and generalizes well between different datasets. Their method was inspired by [59], which uses topological information in random forests for semantic segmentation.

We also approach the edge detection problem as a labeling problem. Our CHM is trained to distinguish between "patches centered on an edge pixel" and "patches centered on a non-edge pixel". We will show that CHM achieves near state-of-the-art performance on the Berkeley dataset [19] and outperforms state-of-the-art methods [11], [12] on NYU depth dataset. Moreover, we will demonstrate that generalization performance of CHM across different datasets is better compared to [11], [12].

3 Contextual Hierarchical Model

The contextual hierarchical model is illustrated in Fig. 2. First, a multi-resolution representation of the input image is obtained by applying downsampling sequentially. Next, a series of classifiers are trained at different resolutions from the finest resolution to the coarsest resolution. At each resolution, the classifier is trained based on the outputs of the previous classifiers in the hierarchy and the input image at that resolution. Finally, the outputs of these classifiers are used to train a new classifier at original resolution. This classifier exploits the rich contextual information from multiple resolutions. The whole training process targets a joint posterior probability at multiple resolutions (see Section 3.3). We describe different steps of the model separately in the following subsections.

3.1 Bottom-Up Step

Let $X = (x(m,n))$ be the $2D$ input image with a corresponding ground truth $Y = (y(m,n))$ where $y(m,n) \in \{0,1\}$ is the class label for pixel (m,n). For notational simplicity, we use $1D$ vectors $X = (x_1, x_2, \ldots, x_n)$ and $Y = (y_1, y_2, \ldots, y_n)$ to denote the input image and corresponding ground truth, respectively[1]. The training dataset then contains K input

1. For notational simplicity we do not use features in out notations. The details about features can be found in Section 3.5.

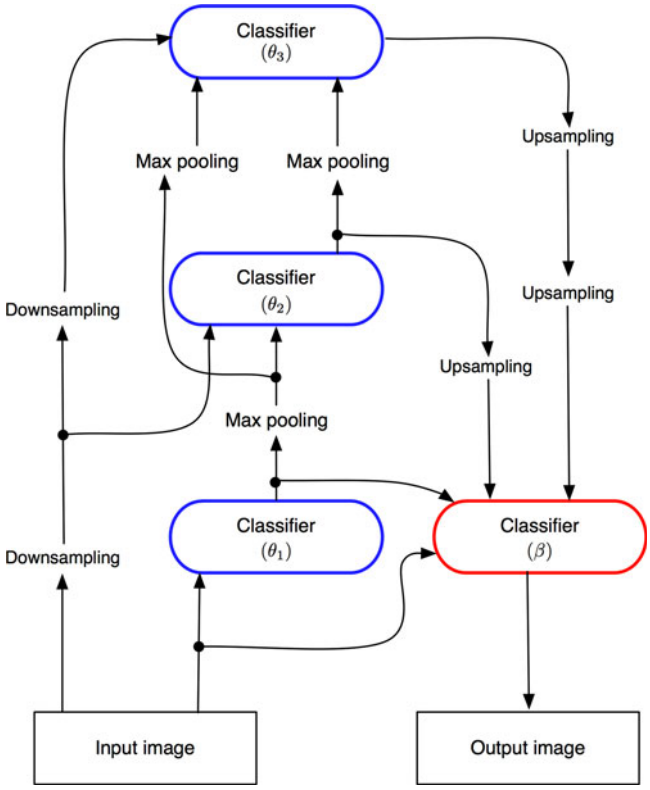

Fig. 2. Illustration of the contextual hierarchical model. The blue classifiers are learned during the bottom-up step and the red classifier is learned during the top-down step. In the bottom-up step, each classifier takes the outputs of lower classifiers as well as the input image as input. The height of the hierarchy, L, is three in this model but it can be extended to any arbitrary number.

images, $\mathbf{X} = \{X_1, X_2, \ldots, X_K\}$, and corresponding ground truth images, $\mathbf{Y} = \{Y_1, Y_2, \ldots, Y_K\}$[2]. We also define the $\Phi(\cdot, l)$ operator which performs down-sampling by a factor of l by averaging the pixels in each 2×2 window, and the $\Gamma(\cdot, l)$ operator which performs max-pooling by a factor of l by finding the maximum pixel value in each 2×2 window. Each classifier in the hierarchy has some internal parameters θ_l, which are learned during training

$$\hat{\theta}_{\mathbf{l}} = \arg\max_{\theta_{\mathbf{l}}} P(\mathbf{\Gamma}(\mathbf{Y}, l-1) \mid \mathbf{\Phi}(\mathbf{X}, l-1), \Gamma(\hat{\mathbf{Y}}^1, l-1), \ldots, \Gamma(\hat{\mathbf{Y}}^{l-1}, 1); \theta_{\mathbf{l}}), \tag{1}$$

where $\hat{\mathbf{Y}}^1, \ldots, \hat{\mathbf{Y}}^{l-1}$ are the outputs of classifiers at the lower levels of the hierarchy. The classifier output of each level is obtained using inference

$$\hat{Y}^l = \arg\max_{Y} P(Y \mid \Phi(X, l-1), \Gamma(\hat{Y}^1, l-1), \ldots, \Gamma(\hat{Y}^{l-1}, 1); \hat{\theta}_{\mathbf{l}}). \tag{2}$$

Each classifier in the l'th level of the hierarchy takes outputs of all lower level classifiers, i.e., $\hat{Y}^1, \ldots, \hat{Y}^{l-1}$, which provide multi-resolution contextual information. For $l = 1$ no prior information is used and the classifier parameters, θ_1, are learned only based on the input image.

2. Unless specified otherwise, upper case symbols, *e.g.*, X, Y, denote a particular vector, lower case symbols, e.g., x, y, denote the elements of a vector, and bold-face symbols, *e.g.*, $\mathbf{X}$, $\mathbf{Y}$, denote a set of vectors.

It is worth mentioning that classifiers at higher levels of the hierarchy have access to contextual information from larger areas because they are trained on down-sampled images.

3.2 Top-Down Step

Unlike the bottom-up step where multiple classifiers are learned, only one classifier is trained in the top-down step. Once all the classifiers are learned in the bottom-up step, a top-down path is used to feed coarser resolution contextual information into a classifier, which is trained at the finest resolution. We define $\Omega(\cdot, l)$ operator that performs up-sampling by a factor of l by duplicating each pixel. For a hierarchical model with L levels, the classifier is trained based on the input image and the outputs of stages 1 to L obtained in the bottom-up step. The internal parameters of the classifier, β, are learned using the following:

$$\hat{\beta} = \arg\max_{\beta} P(\mathbf{Y} \mid \mathbf{X}, \hat{\mathbf{Y}}^1, \Omega(\hat{\mathbf{Y}}^2, 1), \ldots, \Omega(\hat{\mathbf{Y}}^{\mathbf{L}}, L-1); \beta). \tag{3}$$

The output of this classifier can be obtained using the following for inference:

$$\hat{Z} = \arg\max_{Y} P(Y \mid X, \hat{Y}^1, \Omega(\hat{Y}^2, 1), \ldots, \Omega(\hat{Y}^L, L-1); \hat{\beta}). \tag{4}$$

The top-down classifier takes advantage of prior information from multiple resolutions. This multi-resolution prior is an efficient mixture of both local and global information since it is drawn from different scales. In a related work, Seyedhosseini et al. [50] proposed a multi-scale contextual model that exploits contextual information from multiple scales. The advantage of the model proposed here is that the context images are learned at different scales in a supervised framework while the multi-scale contextual model uses simple filtering to create context images at different scales. This allows CHM to optimize a joint posterior at different scales. The overall learning and inference algorithms for the contextual hierarchical model are described in Algorithm 1 and 2, respectively.

Algorithm 1. Learning Algorithm for the CHM

Input: A set of training images together with their binary groundtruth images, $\mathbf{S} = \{(X_i, Y_i), i = 1, \ldots, K\}$ and the height of the hierarchy, L.

Output: $\Theta = \{\hat{\theta}_1, \ldots, \hat{\theta}_L, \hat{\beta}\}$.

- Learn the first classifier, θ_1, using eq. (1) without any prior information and only based on the input image features.
- Compute the output of first classifier, $\hat{\mathbf{Y}}^1$, using equation (2).

for $l = 2$ *to* L **do**

- Learn the l'th classifier, $\hat{\theta}_l$, using equation (1).
- Compute output of the l'th classifier, $\hat{\mathbf{Y}}^l$, using equation (2).

end for

- Learn the top-down classifier, $\hat{\beta}$, using eq. (3).

Algorithm 2. Inference Algorithm for the CHM

Input: An input image X, Θ, L.
Output: $\hat{Z}$.

- Compute the output of first classifier, $\hat{Y}^1$, using equation (2).

for $l = 2$ *to* L **do**

- Compute output of the l'th bottom-up classifier, $\hat{Y}^l$, using eq. (2).

end for

- Compute output of the top-down classifier, $\hat{Z}$, using eq. (4).

3.3 Probabilistic Interpretation

Given the training set $\mathbf{X}$, containing $T = K \times n$ samples, and corresponding labels $\mathbf{Y}$, a common approach is to find the optimal solution by solving the MAP equation

$$\log \prod_t P(Y_t \mid X_t; \Theta). \tag{5}$$

There are two common strategies to solve this optimization. The first strategy, i.e., generative approach, decomposes the posterior to likelihood, $P(X_t \mid Y_t)$, and prior, $P(Y_t)$. The second strategy, i.e., discriminative approach, targets the posterior distribution directly. Our hierarchical model falls into the second category. However, it differs from other approaches in a sense that it optimizes a joint posterior at multiple resolutions, i.e.,

$$\begin{aligned} &\log \prod_t P(Y_t, \Gamma(Y_t,0), \ldots, \Gamma(Y_t, L-1) \mid X_t; \Theta) \\ &\quad = \sum_t \log P(Y_t, \Gamma(Y_t,0), \ldots, \Gamma(Y_t, L-1) \mid X_t; \Theta), \end{aligned} \tag{6}$$

where Γ is the maxpooling operator and L is the number of levels in the hierarchy. This multi-resolution optimization allows us to pool more contextual information from input image. Using $P(A, B \mid C) = P(A \mid B, C)P(B \mid C)$, eq. (6) can be rewritten as

$$\begin{aligned} &\sum_t \log \Big(P\big(Y_t \mid X_t, \Gamma(Y_t,0), \ldots, \Gamma(Y_t, L-1); \Theta\big) \\ &\quad \times P\big(\Gamma(Y_t, L-1) \mid X_t, \Gamma(Y_t,0), \ldots, \Gamma(Y_t, L-2); \Theta\big) \\ &\quad \times \cdots \times P\big(\Gamma(Y_t,0) \mid X_t; \Theta\big) \Big) \\ &= \underbrace{\sum_t \log P\big(Y_t \mid X_t, \Gamma(Y_t,0), \ldots, \Gamma(Y_t, L-1); \Theta\big)}_{\text{Top-down:} J_2(\mathbf{X},\mathbf{Y};\Theta)} \\ &\quad + \underbrace{\sum_l \sum_t \log P\big(\Gamma(Y_t,l) \mid X_t, \Gamma(Y_t,0), \ldots, \Gamma(Y_t, l-1); \Theta\big)}_{\text{Bottom-up:} J_1(\mathbf{X},\mathbf{Y};\Theta)}. \end{aligned} \tag{7}$$

Note that the optimization problems nicely splits down to two subproblems, i.e., $J_1(\mathbf{X}, \mathbf{Y}; \Theta)$ and $J_2(\mathbf{X}, \mathbf{Y}; \Theta)$, which are solved during bottom-up and top-down steps respectively.

In practice, the optimization is done in a greedy way, which means each term in the summation is optimized separately. The output of the classifier at level l, $\hat{Y}^l$, is used as an approximation of the groundtruth at that resolution, $\Gamma(Y, l-1)$. Therefore, the following optimization problems are solved during training

$$\begin{aligned} &\textbf{Bottom-up :} \\ &\max_\Theta J_1(\mathbf{X}, \mathbf{Y}; \Theta) \\ &= \max_\Theta \sum_l \sum_t \log P\big(\Gamma(Y_t, l) \mid X_t, \hat{Y}_t^1, \ldots, \hat{Y}_t^l); \Theta\big) \end{aligned} \tag{8}$$

$$\begin{aligned} &\textbf{Top-down :} \max_\Theta J_2(\mathbf{X}, \mathbf{Y}; \Theta) \\ &= \max_\Theta \sum_t \log P\big(Y_t \mid X_t, \hat{Y}_t^1, \ldots, \hat{Y}_t^L; \Theta\big). \end{aligned} \tag{9}$$

This greedy approach makes the training simple and tractable. It is noteworthy that each of the terms of the outer summation in J_1 is corresponding to one level of the hierarchy. Due to the greedy optimization, a second stage of CHM can improve the results. In the second stage, the top-down classifier of the previous stage is used as the first classifier in the bottom-up step.

3.4 Classifier Selection

Even though our problem formulation is general and not restricted to any specific type of classifier, in practice we need a fast and accurate classifier that is robust to overfitting. Among off-the-shelf classifiers, we consider ANNs, support vector machines (SVM), and random forests. ANNs are slow at training time due to the computational cost of backpropagation. SVMs offer good generalization performance, but choosing the kernel function and the kernel parameters can be time consuming since they need to be adopted for each classifier in the CHM. Furthermore, SVMs are not intrinsically probabilistic and thus are not completely suitable for our CHM model. Random forests provide an unbiased estimate of testing error, but our experiments show that they are prone to overfitting for noisy data. In Section 4.1.1 we show that overfitting can disrupt learning in the CHM model.

We adopt logistic disjunctive normal networks (LDNN) [24] as the classifier in CHM. LDNN is a powerful classifier, which consists of one adaptive layer implemented by logistic sigmoid functions followed by two fixed layers of logical units that compute conjunctions and disjunctions, respectively. LDNN allows an intuitive initialization using k-means clustering and outperforms neural networks, SVMs, and random forests on several standard datasets [24]. Finally, LDNNs are fast to train due to the single adaptive layer, which makes them suitable for the CHM architecture. The details of LDNN can be found in the supplementary materials.

3.5 Feature Selection

In this section, we describe the set of features extracted from input and context images in CHM. The features that we extract from input images include Haar features [60] and histogram of oriented gradients (HOG) features [61]. These features are efficient to compute and somewhat complementary to each other [3]. For color images, Haar and HOG features are computed for each channel separately. We also use dense SIFT features [62] computed at each pixel. In addition, we apply a set of Gabor filters with different parameters and Canny edge detector to obtain more

TABLE 1
Testing Performance of Different Methods on the Weizmann Horse Dataset

Method	F-value	G-mean	Pixel accuracy
KSSVM [14]	?	?	94.60%
TWM [15]	?	?	94.70%
Auto-context [3]	84%	?	?
Levin & Weiss [65]	?	?	95.2%
MSANN [50]	87.58%	92.76%	94.34%
HGM [66]	?	?	**95.9**%
CHM-RF	83.15%	90.20%	92.33%
CHM-LDNN	**89.89**%	**94.39**%	95.37%

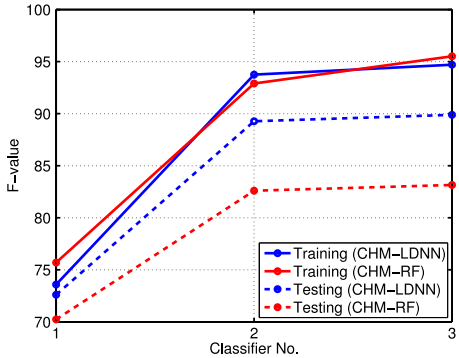

Fig. 3. F-value of the classifiers trained at the original resolution in the CHM with LDNN and random forest. The overfitting in the random forest makes it useless in the CHM architecture.

features. Beside these appearance features, we also use position and its higher orders (up to second order), which are known to be informative for semantic segmentation [16], [35]. These contain the normalized coordinates of each pixel with respect to a certain reference and all the possible multiplications of them. Finally, we use a 15×15 sparse stencil structure [7], which contains 57 samples, to sample the neighborhood around each pixel. In summary, we extract 647 features from color images and 457 features from gray scale images.

Context features are obtained from the outputs of classifiers in the hierarchy. We used a 15×15 stencil to sample context images around each pixel. We also tried larger and more dense sampling structures, e.g., 21×21 patch, but they had negligible impact on the performance. We do not extract any other features beside the neighborhood samples from context images.

4 Experimental Results

We perform experimental studies to evaluate the performance of CHM on three different applications: Semantic segmentation, edge detection, and biomedical image segmentation. The diversity among these applications shows the broad applicability of our method. In all the applications, we used a set of nearly identical parameters, including the number of levels in CHM and the features parameters. Following the reproducible research instructions [63], we maintain a web page containing the source codes and scripts used to generate the results in this section[3].

4.1 Semantic Segmentation

We show the performance of CHM on a binary semantic segmentation dataset, i.e., Weizmann dataset [17], as well as an outdoor scene labeling dataset with multiple classes, i.e., Stanford background dataset [18].

4.1.1 Weizmann Dataset

The Weizmann dataset [17] contains 328 gray scale horse images with corresponding foreground/background truth maps. Similar to Tu and Bai [3], we used half of the images for training and the remaining images were used for testing. The task is to segment horses in each image. We used the features described in Section 3.5. Note that we do not use location information for this dataset since horses are mostly centered in the images, which would create an unfair advantage.

We used a 24×24 LDNN as the classifier in a CHM with two stages and five levels per stage. To improve the generalization performance, we adopted the dropout idea. Hinton et al. [64] showed that removing 50 percent of the hidden nodes in a neural network during the training can improve the performance on the test data. Using the same idea, we randomly removed half of the nodes in the second layer and half of the nodes per group in the first layer at each iteration during the training. At test time, we used the LDNN that contains all of the nodes with their outputs square rooted to compensate for the fact that half of them were active during the training time.

For comparison, we trained a CHM with random forest as the classifier. To avoid overfitting, only $\frac{1}{20}$ of samples were used to train 100 trees in the random forest. We tried different settings for the random forest and picked the best set of parameters. We also trained a multi-scale series of artificial neural networks (MSANN) as in [50]. Three metrics were used to evaluate the segmentation accuracy: Pixel accuracy, F-value $= \frac{2 \times precision \times recall}{precision + recall}$, and G-mean $= \sqrt{recall \times TNR}$ where $TNR = \frac{true\ negative}{true\ negative + false\ positive}$. Unlike F-value, G-mean is symmetric with respect to positive and negative classes. In Table 1 we compare the performance of CHM with some state-of-the-art methods.

CHM outperforms other state-of-the-art methods. It is worth noting that CHM does not make use of fragments and it is based purely on discriminative classifiers that use neighborhood information.

The CHM-LDNN performs at par with the state-of-the-art methods, while the CHM-RF performs worse. The training and testing F-value of the classifiers trained at the original resolution in the CHM, i.e., the classifiers at the bottom of hierarchy, for both LDNN and random forest are shown in Fig. 3. It shows how overfitting propagates through the stages of the CHM when the random forest is used as the classifier. The overfitting disrupts the learning process because there are too few mistakes in the training set compared to the testing set as we go through the stages. For example, the overfitting in the first stage does not permit the second stage to learn the typical mistakes from the first stage that will be encountered at testing time. We tried

3. http://www.sci.utah.edu/~mseyed/Mojtaba_Seyedhosseini/CHM.html

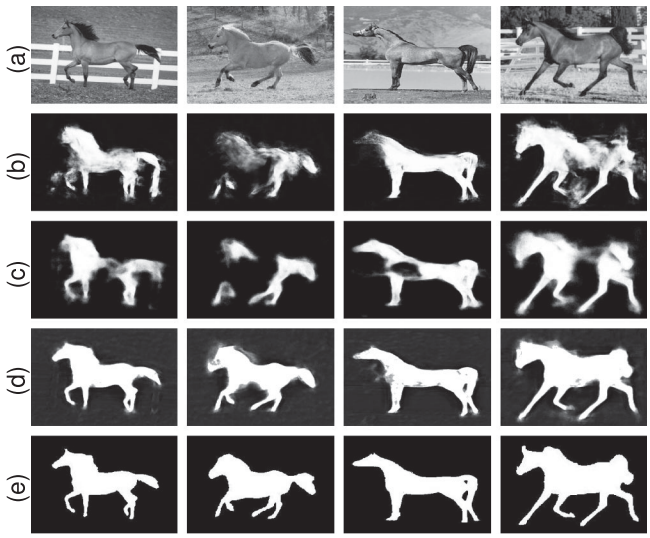

Fig. 4. Test results of the Weizmann horse dataset. (a) Input image, (b) MSANN [50], (c) CHM-RF, (d) CHM-LDNN, (e) ground truth images. The CHM-LDNN is more successful in completing the body of horses.

TABLE 2
Testing Performance of Different Methods on Stanford Background Dataset [18]: Pixelwise Accuracy, Class-Average Accuracy, and Computation Time

Method	Pixel Acc.	Class Acc.	CT (sec.)
Region-based Energy [18]	76.4%	?	10-600
Selecting Regions [26]	79.4%	?	600
Stacked Hierarchical Labeling [35]	76.9%	66.2%	12
Superparsing [67]	77.5%	?	10
Recursive Neural Networks [68]	78.1%	?	?
Pylon Model [69]	81.9%	72.4%	60
Ren et al. [16]	**82.9**%	74.5%	?
Singlescale ConvNet [2]	66%	56.5%	**0.35**
Multiscale ConvNet [2]	78.8%	72.4%	0.6
Multiscale ConvNet+ CRF on gPb [2]	81.4%	**76.0**%	60.5
Series-LDNN	76.35%	72.41%	110
CHM	82.30%	73.70%	60
CHM with Intra-class Connection	**82.95**%	74.32%	65

random forests with different parameters to overcome this problem but were unsuccessful.

Fig. 4 shows four examples of our test images and their segmentation results using different methods. The CHM-LDNN outperforms the other methods in filling the body of horses.

4.1.2 Stanford Background Dataset

The Stanford background dataset [18] contains 715 images of urban and rural scenes, collected from other public datasets such that each image is approximately 240×320 pixels and contains at least one foreground object. This dataset is composed of eight classes, one foreground and seven other classes, and the groundtruth images, obtained from Amazon Mechanical Turk, are included in the dataset. We followed the standard evaluation procedure for this dataset, which is performing five-fold cross-validation with the dataset randomly split into 572 training images and 143 test images.

We trained eight CHMs in a one-versus-all architecture. This is due to our classifier selection, which handles binary classification. To take advantage of intra-class contextual information, we allowed CHMs to communicate with each other at three upper levels of the hierarchy. At those levels, classifiers get samples of context images of other classes as well as their own class. Thus, the feature vector for each class is concatenation of features from all the classes at lower levels. The performance of CHM with and without intra-class connection is reported in Table 2.

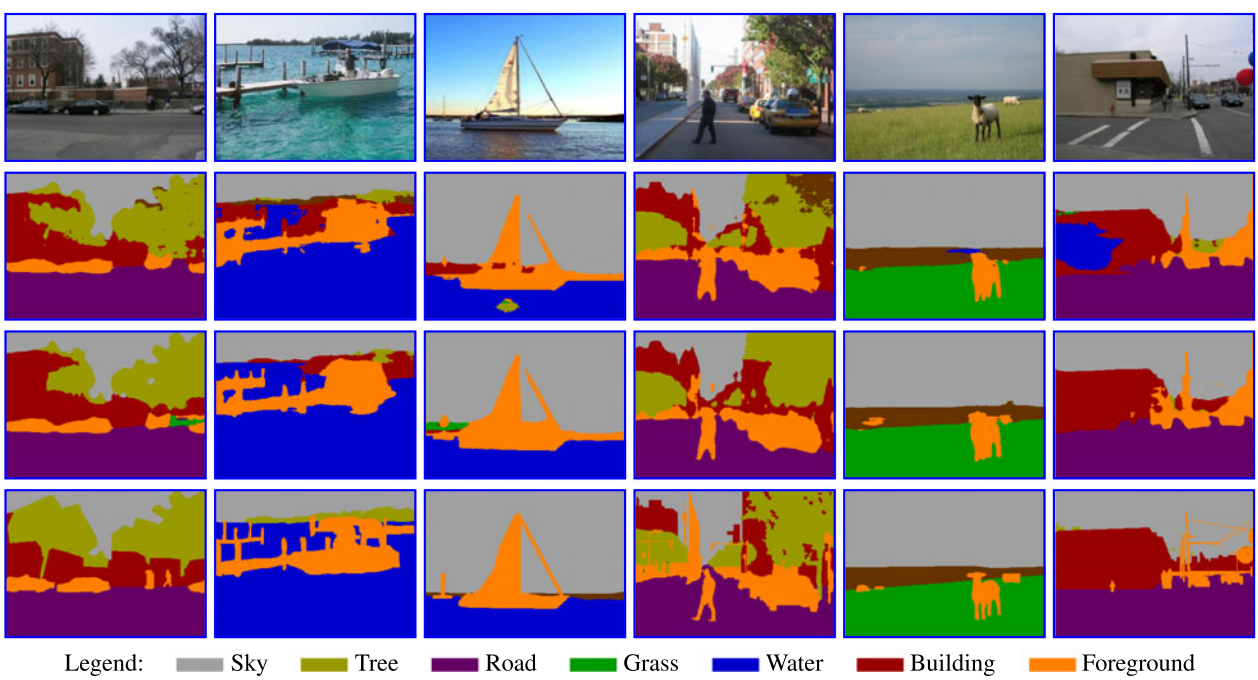

Fig. 5. Test samples of semantic segmentation on Stanford background dataset [18]. *First row:* Input image. *Second row:* CHM. *Third row:* CHM with intra-class connection. *Fourth row:* Groundtruth. Using intra-class contextual information improves the performance.

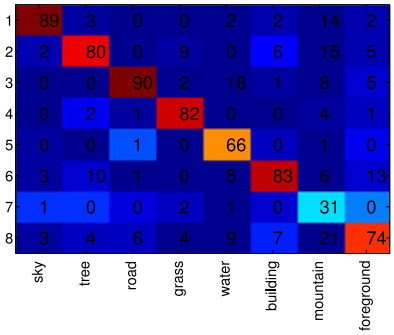

Fig. 6. The confusion matrix of CHM results on the Stanford background dataset [18]. The overall class-average accuracy is 74.32%.

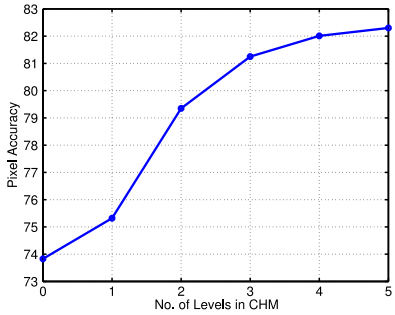

Fig. 7. Performance of CHM on the Stanford Background dataset using different number of levels.

Our CHM achieves state-of-the-art performance in terms of pixel accuracy. Due to the absence of any global constraint for label consistency, CHM performs worse than [2], [16] in terms of class-average accuracy. Similar to [2], we computed superpixels [70] for each image and then assign the most common label, based on CHM output, to each superpixel. Unlike [2], this approach had negligible impact on the performance and improved the pixel accuracy only to 83 percent. This shows CHM is a powerful pixel classifier. In our experiment, inference took about 65 seconds for each image (half of it was spent on computing the features).

A few test samples of the Stanford background dataset and corresponding CHM results are shown in Fig. 5. Using intra-class connection improves the label consistency in the results.

The confusion matrix of CHM is shown in Fig. 6. The hard classes are mountain, water, and foreground. This is consistent with the reported results in [16], [35]. Even though the performance of CHM is similar to [16] for most of the classes, it performs significantly better on the foreground category compared to [16] achieving 74.1 versus 63 percent. We also ran a series architecture with LDNN as classifier to show the effectiveness of our hierarchical model. There were five stages in the series and we used the same set of features as in CHM. The performance was about 6 percent worse than CHM, which asserts the importance of the hierarchy. Finally, we analyzed the effect of different number of levels in CHM. Fig. 7 shows the performance of CHM with different number of levels. It's worth mentioning that the number of levels is limited by the size of image as the the size of image decreases by a factor of four at each level.

4.1.3 SIFT Flow Dataset

The SIFT flow dataset [71] contains 2,488 training and 200 test images. We used the standard split as in [2], [27]. There are 33 classes in this dataset, though, only 30 of them appear in the test set. We trained a similar CHM as in the previous section on this dataset. The performance of CHM for each class in comparison with [27], [72] is depicted in Fig. 8. While the CHM outperforms [27], it performs similar to [72]. Per pixel accuracy and class accuracy of different methods are reported in Table 3. Generally, the CHM performs worse on segmenting more frequent classes such as sky and building, but it performs better on less frequent classes such as bird, streetlight and Balcony. This might be due to the imbalance nature of this dataset.

4.2 Edge Detection

In this section we show the performance of CHM on two edge detection datasets: BSDS 500 [19] and NYU Depth (v2) [20]. We used the popular evaluation framework available in the gPb package [53] to compare CHM performance with other methods. The evaluation framework computes three metrics: Fvalue computed with a fixed threshold for the entire dataset (ODS), F-value computed with per-image best thresholds (OIS), and the average precision (AP).

We trained a CHM with five levels for both datasets. In addition to our regular model, we adopted a multi-scale

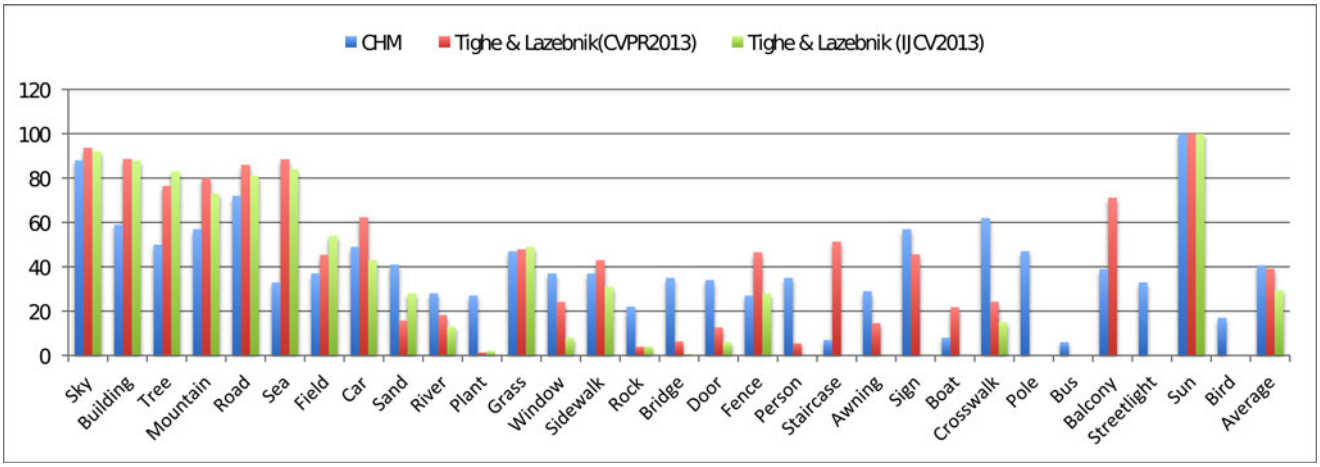

Fig. 8. Per class accuracy of different methods on SIFT flow dataset [71]. The classes are sorted from most frequent to least frequent.

TABLE 3
Testing Performance of Different Methods on the SIFT Flow Dataset

Method	Pixel accuracy	Class accuracy
Tighe and Lazebnik [27]	77.0%	30.1%
Tighe and Lazebnik [72]	78.6%	39.2%
Multiscale ConvNet (natural frequencies) [2]	78.5%	29.6%
Multiscale ConvNet (balanced frequencies) [2]	72.3%	50.8%
Pinheiro and Collobert [73]	77.7%	29.8%
Long et al. [47]	**85.2**%	**51.7**%
CHM	61.68%	40.66%

TABLE 4
Testing Performance of Different Methods on BSDS 500 Dataset [19]

Method	ODS	OIS	AP	CT (sec.)
gPb-OWT-UCM [53]	0.726	0.760	0.727	240
Sketch Tokens [58]	0.728	0.746	0.780	1
SCG [12]	0.739	0.758	0.773	280
SE-SS [11]	0.73	0.75	0.77	**1/30**
CHM-SS	0.722	0.737	0.772	100
SE-MS [11]	**0.741**	**0.760**	0.780	1/6
CHM-MS	0.735	0.751	**0.804**	190

CHM achieves near state-of-the-art performance in terms of ODS and OIS, and improves over other methods significantly in terms of AP. SS:single-scale, MS:multi-scale, CT:computation time.

strategy similar to [11], [58] to compute edge maps. That is, at test time, we ran the trained CHM on the original, as well as double and half resolution versions of each input image. We then resized the results to the original image resolution and averaged them to obtain the edge map. We also used the standard non-maximal suppression, suggested in [11], [12], [53], [58], to obtain thinned edges.

4.2.1 BSDS 500 Dataset

Berkeley segmentation dataset and benchmarks (BSDS 500) [19], [53] is an extension of BSDS 300 dataset and used widely for the evaluation of edge detection techniques. It contains 200 training, 100 validation, and 200 testing images of resolution 321×481 pixels (roughly). The human annotations for each image is included in the dataset. The precision-recall curves for CHM and four other methods are shown in Fig. 9.

Note that CHM achieves high precision and recall at both ends of the precision-recall curve. The evaluation metrics are reported in Table 4.

While CHM performs about the same as SCG [12] and SE [11] in terms od ODS and OIS, it achieves state-of-the-art performance in terms of AP. It must be emphasized that unlike gPb [53] and SCG [12], our CHM does not include any globalization step and only relies on the local patch information. In addition, our CHM is a general patch-based model and unlike gPb [53], SCG [12], and SE [11] can be used in general semantic segmentation frameworks. Finally we will show in Section 4.2.3 that the cross-dataset generalization performance of CHM is significantly better than other learning-based approaches, i.e., sketch tokens [58], SCG [12], and SE [11]. A few test examples of BSDS 500 dataset and corresponding edge detection results are shown in Fig. 10. As shown in our results, CHM captures finer details such as upper stairs in the first row, steeples in the second row, and wheels in the third row.

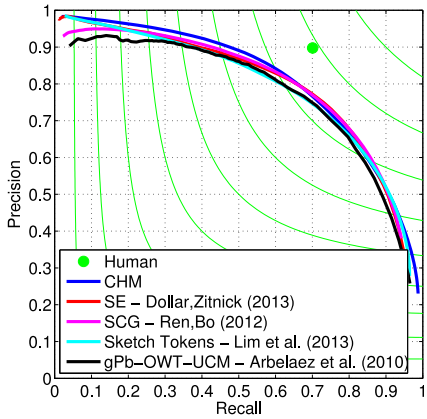

Fig. 9. Precision-recall curves of CHM in comparison with other methods for BSDS 500 dataset [19].

4.2.2 NYU Depth Dataset (v2)

The NYU depth dataset (v2) [20] is an RGB-D dataset containing 1,449 pairs of RGB and depth images of resolution 480×640 pixels, with corresponding groundtruth semantic segmentations. We used the scripts provided by the authors of [12] to adopt this dataset for edge detection[4]. They used 60 percent of the images for training (869 images) and the remaining 40 percent for testing (580 images). The images were also resized to 240×320 resolution. We evaluated the performance of CHM using RGB and RGBD modalities. For the depth channel, we computed the same set of features that we extract from the RGB color channels. In Table 5, we compare CHM with SCG [12] and SE [11].

CHM performs significantly better than other methods and reaches an F-value of 0.649 for RGB and 0.678 for RGBD. Unlike [11], [12], our CHM does not benefit too much from the multi-scale strategy. This can assert that CHM takes advantage of multi-scale information effectively that later multi-scale strategies would have marginal impact. Qualitative comparisons are shown in Fig. 11 and the precision-recall curves are shown in Fig. 12.

4.2.3 Cross-Dataset Generalization

Inspired by the work of Dollár and Zitnick [11], we performed a set of experiments to examine the generalization performance of CHM in comparison to other learning-based methods. We used the trained CHM on BSDS 500 dataset and ran it on NYU depth dataset for RGB modality. The authors of sketch tokens [58], SCG [12], and SE [11] have provided their models for BSDS 500 dataset; so, we could run the same experiment for their methods. The performance metrics for different methods are reported in Table 6 and corresponding precision-recall curves are shown in Fig. 13.

4. The scripts are available at http://homes.cs.washington.edu/~xren/research/nips2012/sparse_contour_gradients_v1.1.zip

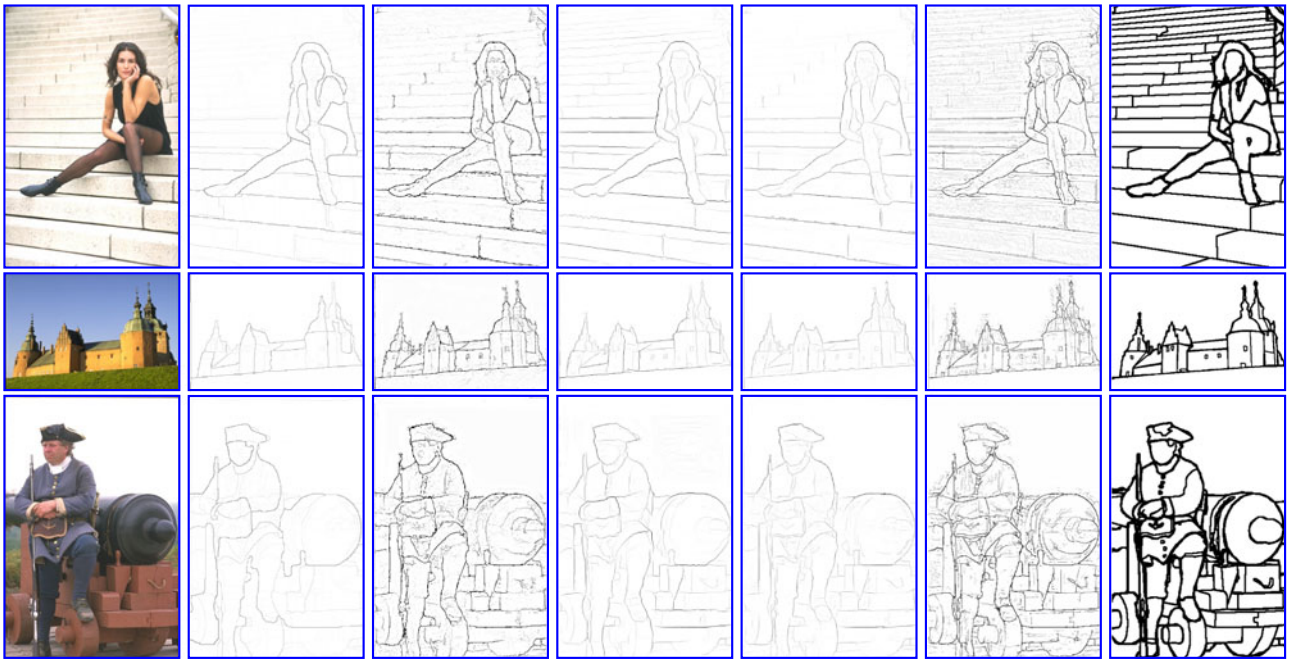

Fig. 10. Test samples of edge detection on BSDS 500 [19] dataset. (a) Input image, (b) gPb-OWT-UCM [53], (c) Sketch tokens [58], (d) SCG [12], (e) SE [11], (f) CHM, (g) Groundtruth. CHM is able to capture finer details like upper stairs in the first row, steeples in the second row, and wheels in the third row.

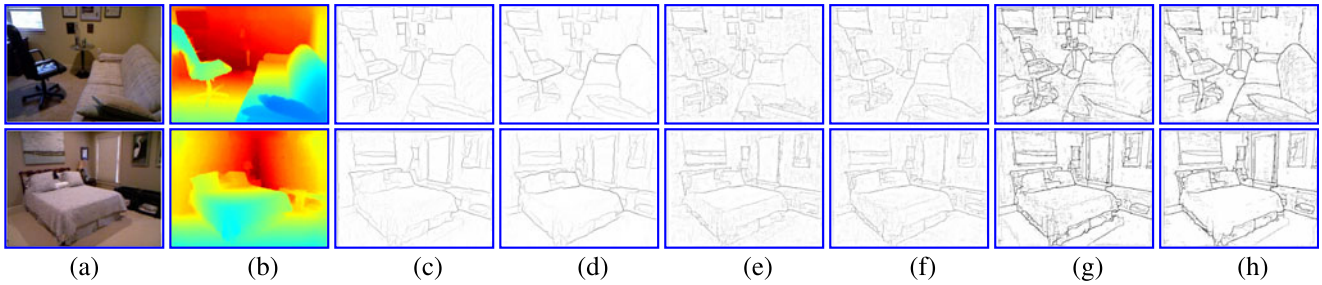

Fig. 11. Test samples of edge detection on NYU depth (v2) dataset [20]. (a) Input image, (b) Depth image, (c) SCG (RGB) [12], (d) SCG (RGBD) [12], (e) SE (RGB) [11], (f) SE (RGBD) [11], (g) CHM (RGB), and (h) CHM (RGBD).

CHM performs significantly better than other methods. Note that all methods perform about the same on BSDS 500 dataset (Table 4). We believe this asserts that our CHM can be used as a general edge detection technique.

TABLE 5
Testing Performance of Different Methods on NYU Depth Dataset [20] Using RGB (Top), and RGBD (Bottom) Modalities

Method	ODS	OIS	AP	CT (sec.)
SCG [12] (RGB)	0.557	0.569	0.438	280
SE-SS [11] (RGB)	0.58	0.59	0.53	**1/30**
CHM-SS (RGB)	0.648	0.658	0.614	50
SE-MS [11] (RGB)	0.596	0.608	0.541	1/6
CHM-MS (RGB)	**0.649**	**0.661**	**0.625**	85
SCG [12] (RGBD)	0.621	0.632	0.534	280
SE-SS [11] (RGBD)	0.62	0.63	0.59	**1/25**
CHM-SS (RGBD)	0.678	0.690	0.665	90
SE-MS [11] (RGBD)	0.636	0.647	0.601	1/5
CHM-MS (RGBD)	**0.678**	**0.690**	**0.665**	120

CHM achieves state-of-the-art performance for both cases. SS: single-scale, MS: multi-scale, CT: computation time.

4.3 Biomedical Image Segmentation

In the last set of experiments, we applied CHM to the membrane detection problem in electron microscopy images. This is a challenging problem because of the noisy texture,

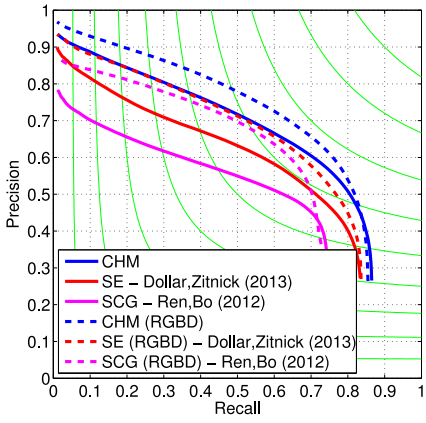

Fig. 12. Precision-recall curves of different methods for NYU depth dataset [20] using RGB (solid lines) and RGBD (dashed lines) modalities.

TABLE 6
Testing Performance of Different Methods on NYU Depth Dataset [20] Using BSDS 500 Dataset [19] for Training

Method	ODS	OIS	AP
Sketch Tokens [58]	0.567	0.581	0.490
SCG [12]	0.568	0.579	0.441
SE [11]	0.552	0.566	0.462
CHM	**0.595**	**0.606**	**0.528**

CHM outperforms other learning-based approaches significantly.

TABLE 7
Testing Performance of Different Methods for the Mouse Neuropil and Drosophila VNC Datasets

	Mouse neuropil		Drosophila VNC	
Method	F-value	G-mean	F-value	G-mean
gPb-OWT-UCM [10]	45.68%	64.75%	49.90%	69.57%
BEL [55]	71.68%	84.46%	70.21%	84.20%
MSANN [50]	81.99%	90.48%	78.89%	88.74%
CHM	**86.00%**	**92.48%**	**80.72%**	**90.02%**

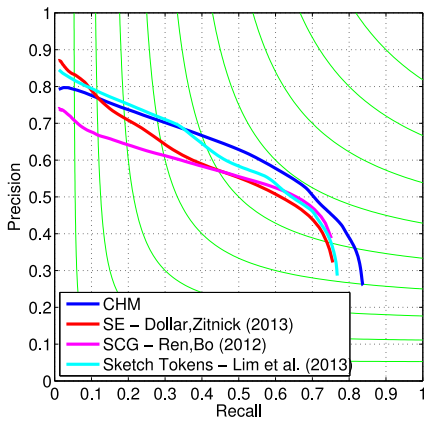

Fig. 13. Precision-recall curves of different methods for NYU depth dataset [20] using BSDS 500 dataset [19] for training. Cross-dataset generalization performance of CHM is better compared to other methods.

complex intracellular structures, and similar local appearances among different objects [42], [74]. In these experiments, we used a CHM with two stages and five levels per stage. A 24×24 LDNN was used as the classifier. In addition to the feature set described in Section 3.5, we included Radon-like features (RLF) [75], which proved to be informative for membrane detection.

4.4 Mouse Neuropil Dataset

This dataset is a stack of 70 images from the mouse neuropil acquired using serial block face scanning electron microscopy (SBFSEM [76]). It has a resolution of $10 \times 10 \times 50$ nm/pixel and each $2D$ image is 700 by 700 pixels. An expert anatomist annotated membranes, i.e., cell boundaries, in these images. From those 70 images, 14 images were randomly selected and used for training and the 56 remaining images were used for testing. The task is to detect membranes in each $2D$ section.

Since the task is detecting the boundary of cells, we compared our method with two general boundary detection methods, gPb-OWT-UCM (global probability of boundary followed by the oriented watershed transform and ultrametric contour maps) [10] and boosted edge learning (BEL) [55]. The testing results for different methods are given in Table 7. The CHM-LDNN outperforms the other methods with a notably large margin.

One example of the test images and corresponding membrane detection results using different methods are shown in Fig. 14. As shown in our results, the CHM outperforms MSANN in removing undesired parts from the background and closing some gaps.

4.5 Drosophila VNC Dataset

This dataset contains 30 images from Drosophila first instar larva ventral nerve cord (VNC) [21], [22] acquired using serial-section transmission electron microscopy [77], [78]. Each image is 512 by 512 pixels and the resolution is $4 \times 4 \times 50$ nm/pixel. The membranes are marked by a human expert in each image. We used 15 images for training and 15 images for testing. The testing performance for different methods are reported in Table 7. It can be seen that the CHM outperforms the other methods in terms of pixel

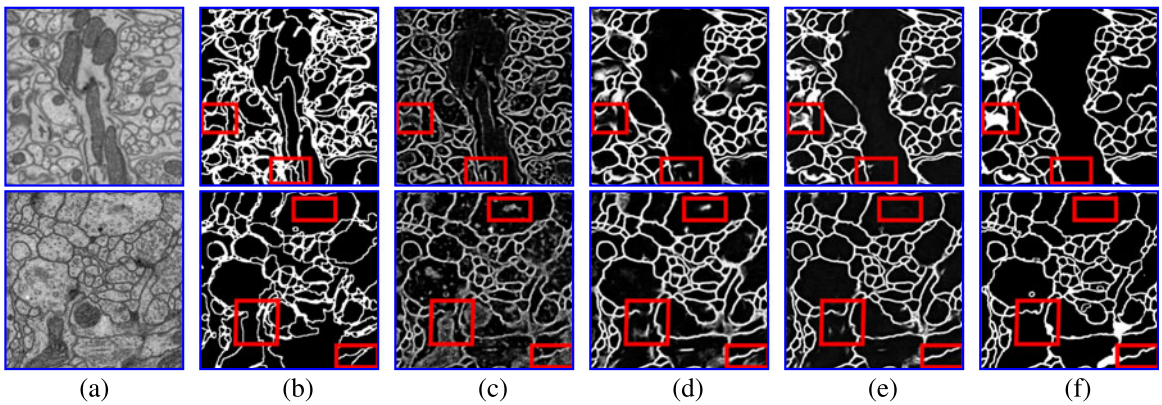

Fig. 14. Test results of the mouse neuropil dataset (first row) and the Drosophila VNC dataset (second row). (a) Input image, (b) gPb-OWT-UCM [10], (c) BEL [55], (d) MSANN [50], (e) CHM-LDNN, (f) ground truth images. The CHM is more successful in removing undesired parts and closing small gaps. Some of the improvements are marked with red rectangles. For gPb-OWT-UCM method, the best threshold was picked and the edges were dilated to the true membrane thickness.

TABLE 8
Pixel Error (1-F-value) and Training Time (Hours) of Different Methods on ISBI Challenge [79] Test Set

Method	$1-$F-value	Training Time
Laptev et al. [80]	0.067	?
Convolutional Networks [42]	0.067	?
Human	0.066	–
Deep Neural Networks [44]	0.065	$85(GPU)$
CHM	**0.063**	$30(CPU)$

Numbers are available on the challenge leader board.

error. One test sample and membrane detection results for different methods are shown in Fig. 14.

The same dataset was used as the training set for the ISBI 2012 EM challenge [79]. The participants were asked to submit the results on a test set (the same size as the training set) to the challenge server. We trained the same model on the whole 30 images and submitted the results for the testing volume to the server. The pixel error (1-F-value) of different methods are reported in Table 8. CHM achieved pixel error of 0.063 which is better than the human error, i.e., how much a second human labeling differed from the first one. It also outperformed the convolutional networks proposed in [42] and [44]. It is noteworthy that CHM is significantly faster than deep neural networks (DNN) [44] at training. While DNN needs 85 hours on GPU for training, CHM only needs 30 hours on CPU. At test time, CHM can be slower due to the feature computation time.

5 Conclusion and Future Work

We develop a discriminative learning scheme for semantic segmentation, called CHM, which takes advantage of contextual information at multiple resolutions in a hierarchy. The main advantage of CHM is its ability to optimize a posterior probability at multiple resolutions. To our knowledge, this is the first time that a posterior at multiple resolutions is optimized for semantic segmentation. CHM performs this optimization efficiently in a greedy manner. To achieve this goal, CHM trains several classifiers at multiple resolutions and leverages the obtained results for learning a classifier at the original resolution. We applied our model to several challenging datasets for semantic segmentation, edge detection, and biomedical image segmentation. Results indicate that CHM achieves state-of-the-art performance on all of these applications.

An important characteristic of CHM is that it is only based on patch information and does not make use of any exemplars or shape models. This enables CHM to serve as a general labeling method with high accuracy. The other advantage of CHM is its simple training. Even though our model needs to learn hundreds of parameters, the training remains tractable since classifiers are trained separately.

We conclude by discussing a possible extension of the CHM. Even though CHM is able to model global contextual information within a scene, it can be prone to error due to absence of any global constrains. Therefore, CHM can be used as a first step in a semantic segmentation pipeline. Postprocessing such as CRF proposed in [2] can be used to enforce label consistency and global constraints

Acknowledgments

This work was supported by NIH 1R01NS075314-01 (TT, MHE) and NSF IIS-1149299(TT). The authors thank the "National Center for Microscopy Imaging Research" and the "Cardona Lab at HHMI Janelia Farm" for providing the mouse neuropil and Drosophila VNC datasets. They also thank Piotr Dollár for providing edge detection results of SE [11] method for NYU depth dataset.

References

[1] S. Z. Li, *Markov Random Field Modeling in Computer Vision*. New York, NY, USA: Springer-Verlag, 1995.
[2] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1915–1929, Aug. 2013.
[3] Z. Tu and X. Bai, "Auto-context and its application to high-level vision tasks and 3d brain image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 10, pp. 1744–1757, Oct. 2010.
[4] A. Torralba, K. P. Murphy, and W. T. Freeman, "Contextual models for object detection using boosted random fields," in *Proc. Adv. Neural Inf. Process. Syst.*, 2004, pp. 1401–1408.
[5] G. Heitz, S. Gould, A. Saxena, and D. Koller, "Cascaded classification models: Combining models for holistic scene understanding," in *Proc. Adv. Neural Inf. Process. Syst.*, 2008, pp. 641–648.
[6] C. Li, A. Kowdle, A. Saxena, and T. Chen, "Toward holistic scene understanding: Feedback enabled cascaded classification models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1394–1408, Jul. 2012.
[7] E. Jurrus, A. R. C. Paiva, S. Watanabe, J. R. Anderson, B. W. Jones, R. T. Whitaker, E. M. Jorgensen, R. E. Marc, and T. Tasdizen, "Detection of neuron membranes in electron microscopy images using a serial neural network architecture," *Med. Image Anal.*, vol. 14, no. 6, pp. 770–783, 2010.
[8] Z. Ren and G. Shakhnarovich, "Image segmentation by cascaded region agglomeration," in *Proc. IEEE Conf. Comput. Vision Pattern. Recog.*, 2013, pp. 2011–2018.
[9] M. Seyedhosseini and T. Tasdizen, "Multi-class multi-scale series contextual model for image segmentation," *IEEE Trans. Image Process.*, vol. 22, no. 11, pp. 4486–4496, Nov. 2013.
[10] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "From contours to regions: An empirical evaluation," in *Proc. IEEE Conf. Comput. Vision Pattern. Recog.*, 2009, pp. 2294–2301.
[11] P. Dollár and C. L. Zitnick, "Structured forests for fast edge detection," in *Proc. IEEE Int. Conf. Comput. Vision*, 2013, pp. 1841–1848.
[12] X. Ren and L. Bo, "Discriminatively trained sparse code gradients for contour detection," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 584–592.
[13] B. Catanzaro, B.-Y. Su, N. Sundaram, Y. Lee, M. Murphy, and K. Keutzer, "Efficient, high-quality image contour detection," in *Proc. IEEE Int. Conf. Comput. Vision*, 2009, pp. 2381–2388.
[14] L. Bertelli, T. Yu, D. Vu, and B. Gokturk, "Kernelized structural SVM learning for supervised object segmentation," in *Proc. IEEE Conf. Comput. Vision Pattern. Recog.*, 2011, pp. 2153–2160.
[15] D. Kuettel and V. Ferrari, "Figure-ground segmentation by transferring window masks," in *Proc. IEEE Conf. Comput. Vision Pattern. Recog.*, 2012, pp. 558–565.
[16] X. Ren, L. Bo, and D. Fox, "RGB-(D) scene labeling: Features and algorithms," in *Proc. IEEE Conf. Comput. Vision Pattern. Recog.*, 2012, pp. 2759–2766.
[17] E. Borenstein, E. Sharon, and S. Ullman, "Combining top-down and bottom-up segmentation," *Proc. Comput. Vision Pattern Recog. Workshop*, 2004, pp. 46–46.
[18] S. Gould, R. Fulton, and D. Koller, "Decomposing a scene into geometric and semantically consistent regions," in *Proc. IEEE Int. Conf. Comput. Vision*, 2009, pp. 1–8.
[19] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. IEEE Int. Conf. Comput. Vision*, 2001, pp. 416–423.
[20] N. Silberman and R. Fergus, "Indoor scene segmentation using a structured light sensor," in *Proc. IEEE Int. Conf. Comput. Vision Workshop*, 2011, pp. 601–608.

[21] A. Cardona, S. Saalfeld, S. Preibisch, B. Schmid, A. Cheng, J. Pulokas, P. Tomančák, and V. Hartenstein, "An integrated micro- and macroarchitectural analysis of the *Drosophila* brain by computer-assisted serial section electron microscopy," *PLoS Biol.*, vol. 8, no. 10, p. e1000502, Oct. 2010.

[22] A. Cardona, S. Saalfeld, J. Schindelin, I. Arganda-Carreras, S. Preibisch, M. Longair, P. Tomancak, V. Hartenstein, and R. J. Douglas, "TrakEM2 software for neural circuit reconstruction," *PLoS ONE*, vol. 7, no. 6, p. e38011, Jun. 2012.

[23] O. Sporns, G. Tononi, and R. Ktter, "The human connectome: A structural description of the human brain," *PLoS Comput. Biol.*, vol. 1, p. e42, 2005.

[24] M. Seyedhosseini, M. Sajjadi, and T. Tasdizen, "Image segmentation with cascaded hierarchical models and logistic disjunctive normal networks," in *Proc. IEEE Int. Conf. Comput. Vision*, 2013, pp. 2168–2175.

[25] D. Larlus and F. Jurie, "Combining appearance models and Markov random fields for category level object segmentation," in *Proc. IEEE Conf. Comput. Vision Pattern. Recog.*, 2008, pp. 1–7.

[26] M. P. Kumar and D. Koller, "Efficiently selecting regions for scene understanding," in *Proc. IEEE Conf. Comput. Vision Pattern. Recog.*, 2010, pp. 3217–3224.

[27] J. Tighe and S. Lazebnik, "Superparsing," *Int. J. Comput. Vision*, vol. 101, no. 2, pp. 329–349, 2013.

[28] X. He, R. Zemel, and M. Carreira-Perpinan, "Multiscale conditional random fields for image labeling," in *Proc. IEEE Conf. Comput. Vision Pattern. Recog.*, 2004, pp. II-695–II-702.

[29] L. Ladicky, C. Russell, P. Kohli, and P. H. Torr, "Associative hierarchical CRFs for object class image segmentation," in *Proc. IEEE Int. Conf. Comput. Vision*, 2009, pp. 739–746.

[30] V. Koltun, "Efficient inference in fully connected CRFs with Gaussian edge potentials," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 109–117.

[31] J. Yao, S. Fidler, and R. Urtasun, "Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation," in *Proc. IEEE Conf. Comput. Vision Pattern. Recog.*, 2012, pp. 702–709.

[32] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 9, pp. 1124–1137, Sep . 2004.

[33] P. Kohli, L. Ladický, and P. H. Torr, "Robust higher order potentials for enforcing label consistency," *Int. J. Comput. Vision*, vol. 82, no. 3, pp. 302–324, 2009.

[34] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.

[35] D. Munoz, J. A. Bagnell, and M. Hebert, "Stacked hierarchical labeling," in *Proc. Eur. Conf. Comput. Vision*, 2010, pp. 57–70.

[36] D. Grangier, L. Bottou, and R. Collobert, "ICML 2009 Deep Learning Workshop," in *Proc. Int. Conf. Mach. Learning*, 2009.

[37] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," in *Intelligent Signal Processing*. Piscataway, NJ, USA: IEEE Press, 2001, pp. 306–351.

[38] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[39] D. Ciresan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *Proc. IEEE Conf. Comput. Vision Pattern. Recog.*,2012, pp. 3642–3649.

[40] C. Szegedy, A. Toshev, and D. Erhan, "Deep neural networks for object detection," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 2553–2561.

[41] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun, "What is the best multi-stage architecture for object recognition?" in *Proc. IEEE Int. Conf. Comput. Vision*, 2009, pp. 2146–2153.

[42] V. Jain, J. F. Murray, F. Roth, S. Turaga, V. Zhigulin, K. L. Briggman, M. N. Helmstaedter, W. Denk, and H. S. Seung, "Supervised learning of image restoration with convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vision*, 2007, pp. 1–8.

[43] S. C. Turaga, K. L. Briggman, M. Helmstaedter, W. Denk, and H. S. Seung, "Maximin affinity learning of image segmentation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 1865–1873.

[44] D. Ciresan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, "Deep neural networks segment neuronal membranes in electron microscopy images," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 2852–2860.

[45] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. Torr, "Conditional random fields as recurrent neural networks," *arXiv preprint arXiv:1502.03240*, 2015.

[46] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected CRFs," arXiv preprint arXiv:1412.7062, 2014.

[47] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vision Pattern. Recog.*, 2015.

[48] M. Fink and P. Perona, "Mutual boosting for contextual inference," in *Proc. Adv. Neural Inf. Process. Syst.*, 2004, pp. 9 –16.

[49] S. Haykin, *Neural Networks—A Comprehensive Foundation*, 2nd ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 1999.

[50] M. Seyedhosseini, R. Kumar, E. Jurrus, R. Guily, M. Ellisman, H. Pfister, and T. Tasdizen, "Detection of neuron membranes in electron microscopy images using multi-scale context and radon-like features," *Med. Image Comput. Comput. Assist. Interv.*, vol. 14, pp. 670–677, 2011.

[51] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.

[52] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 7, pp. 629–639, Jul. 1990.

[53] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, May 2011.

[54] D. R. Martin, C. C. Fowlkes, and J. Malik, "Learning to detect natural image boundaries using local brightness, color, and texture cues," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 5, pp. 530–549, May 2004.

[55] P. Dollár, Z. Tu, and S. Belongie, "Supervised learning of edges and object boundaries," in *Proc. IEEE Conf. Comput. Vision Pattern. Recog.*, 2006, pp. 1964–1971.

[56] J. Mairal, M. Leordeanu, F. Bach, M. Hebert, and J. Ponce, "Discriminative sparse image models for class-specific edge detection and image interpretation," in *Proc. Eur. Conf. Comput. Vision*, 2008, pp. 43–56.

[57] S. Gupta, P. Arbelaez, and J. Malik, "Perceptual organization and recognition of indoor scenes from RGB-D images," in *Proc. IEEE Conf. Comput. Vision Pattern. Recog.*, 2013, pp. 564–571.

[58] J. J. Lim, C. L. Zitnick, and P. Dollár, "Sketch tokens: A learned mid-level representation for contour and object detection," in *Proc. IEEE Conf. Comput. Vision Pattern. Recog.*, 2013, pp. 3158–3165.

[59] P. Kontschieder, S. R. Bulo, H. Bischof, and M. Pelillo, "Structured class-labels in random forests for semantic image labelling," in *Proc. IEEE Int. Conf. Comput. Vision*, 2011, pp. 2190–2197.

[60] P. Viola and M. J. Jones, "Robust real-time face detection," *Int. J. Comput. Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[61] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Conf. Comput. Vision Pattern. Recog.*, 2005, pp. 886–893.

[62] C. Liu, J. Yuen, and A. Torralba, "SIFT Flow: Dense correspondence across scenes and its applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 978–994, May 2011.

[63] P. Vandewalle, J. Kovacevic, and M. Vetterli. Reproducible research. http://reproducibleresearch.net/index.php/Main_Page, 2006.

[64] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[65] A. Levin and Y. Weiss, "Learning to combine bottom-up and top-down segmentation," in *Proc. Eur. Conf. Comput. Vision*, 2006, pp. 581–594.

[66] G. Liu, Z. Lin, X. Tang, and Y. Yu, "A hybrid graph model for unsupervised object segmentation," in *Proc. IEEE Int. Conf. Comput. Vision*, 2007, pp. 1–8.

[67] J. Tighe and S. Lazebnik, "Superparsing: Scalable nonparametric image parsing with superpixels," in *Proc. Eur. Conf. Comput. Vision*, 2010, pp. 352–365.

[68] R. Socher, C. C. Lin, A. Ng, and C. Manning, "Parsing natural scenes and natural language with recursive neural networks," in *Proc. Int. Conf. Mach. Learning*, 2011, pp. 129–136.

[69] V. Lempitsky, A. Vedaldi, and A. Zisserman, "A pylon model for semantic segmentation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 1485–1493.
[70] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. Comput. Vision*, vol. 59, no. 2, pp. 167–181, 2004.
[71] C. Liu, J. Yuen, and A. Torralba, "Nonparametric scene parsing via label transfer," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 12, pp. 2368–2382, Dec. 2011.
[72] J. Tighe and S. Lazebnik, "Finding things: Image parsing with regions and per-exemplar detectors," in *Proc. IEEE Conf. Comput. Vision Pattern. Recog.*, 2013, pp. 3001–3008.
[73] P. Pinheiro and R. Collobert, "Recurrent convolutional neural networks for scene labeling," in *Proc. Int. Conf. Mach. Learning*, 2014, pp. 82–90.
[74] A. Lucchi, K. Smith, R. Achanta, V. Lepetit, and P. Fua, "A fully automated approach to segmentation of irregularly shaped cellular structures in EM images," *Med. Image Comput. Comput. Assist. Interv.*, vol. 13, pp. 463–471, 2010.
[75] R. Kumar, A. Vázquez-Reina, and H. Pfister, "Radon-like features and their application to connectomics," in *Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recog. Workshop*, Jun. 2010, pp. 186–193.
[76] W. Denk and H. Horstmann, "Serial block-face scanning electron microscopy to reconstruct three-dimensional tissue nanostructure," *PLoS Biol.*, vol. 2, p. e329, 2004.
[77] J. R. Anderson, B. W. Jones, J.-H. Yang, M. V. Shaw, C. B. Watt, P. Koshevoy, J. Spaltenstein, E. Jurrus, K. Uv, R. T. Whitaker, D. Mastronarde, T. Tasdizen, and R. E. Marc, "A computational framework for ultrastructural mapping of neural circuitry," *PLoS Biol.*, vol. 7, no. 3, p. e1000074, Mar. 2009.
[78] D. B. Chklovskii, S. Vitaladevuni, and L. K. Scheffer, "Semi-automated reconstruction of neural circuits using electron microscopy," *Current Opinion Neurobiol.*, vol. 20, no. 5, pp. 667–675, 2010.
[79] I. Arganda-Carreras, S. Seung, A. Cardona, and J. Schindelin. (2012). ISBI2012 segmentation of neuronal structures in EM stacks. [Online]. Available: http://brainiac2.mit.edu/isbi_challenge/
[80] D. Laptev, A. Vezhnevets, S. Dwivedi, and J. Buhmann, "Anisotropic system image segmentation using dense correspondence across sections," in *Proc. Med. Image Comput. Comput. Assist. Interv.*, 2012, pp. 323–330.

Mojtaba Seyedhosseini received the B.S. degree in Electrical Engineering from the University of Tehran in 2007, and the M.S. degree in Electrical Engineering from the Sharif University of Technology in 2009. He got his PhD from the Scientific Computing and Imaging (SCI) Institute at the University of Utah in 2014. He is now with Google Inc. His research interests include computer vision, machine learning, statistical pattern recognition, and image analysis.

Tolga Tasdizen received the B.S. degree in electrical and electronics engineering from Bogazici University in 1995. He received his M.S. and Ph. D. degrees in engineering from Brown University in 1997 and 2001, respectively. He held Postdoctoral researcher and Research Assistant Professor positions with the Scientific Computing and Imaging Institute, University of Utah, Salt Lake City, UT, USA 2001-4 and 2004-8, respectively. Since 2008, he has been with the Department of Electrical and Computer Engineering at the University of Utah where he is currently an Associate Professor. Dr. Tasdizen is also a Utah Science Technology and Research Initiative (USTAR) faculty member in the SCI Institute. His research interests are in image processing, computer vision and pattern recognition with a focus on applications in biological and medical image analysis. Dr. Tasdizen is a recipient of the National Science Foundation's CAREER award. He is a member of Bio Imaging and Signal Processing Technical Committee (BISP TC) of the IEEE Signal Processing Society and serves as an associate editor for the IEEE Signal Processing Letters and BMC Bioinformatics.